package com.twitter.cr\_mixer.module

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.simclusters\_v2.common.SimClustersEmbedding

import com.twitter.simclusters\_v2.common.TweetId

import com.twitter.simclusters\_v2.common.UserId

import com.twitter.storehaus.ReadableStore

import com.twitter.strato.client.{Client => StratoClient}

import com.twitter.representation\_manager.thriftscala.SimClustersEmbeddingView

import com.twitter.simclusters\_v2.thriftscala.EmbeddingType

import com.twitter.simclusters\_v2.thriftscala.ModelVersion

import com.google.inject.Provides

import com.google.inject.Singleton

import javax.inject.Named

import com.twitter.cr\_mixer.model.ModuleNames

import com.twitter.frigate.common.store.strato.StratoFetchableStore

import com.twitter.hermit.store.common.ObservedReadableStore

import com.twitter.simclusters\_v2.thriftscala.{SimClustersEmbedding => ThriftSimClustersEmbedding}

object RepresentationManagerModule extends TwitterModule {

private val ColPathPrefix = "recommendations/representation\_manager/"

private val SimclustersTweetColPath = ColPathPrefix + "simClustersEmbedding.Tweet"

private val SimclustersUserColPath = ColPathPrefix + "simClustersEmbedding.User"

@Provides

@Singleton

@Named(ModuleNames.RmsTweetLogFavLongestL2EmbeddingStore)

def providesRepresentationManagerTweetStore(

statsReceiver: StatsReceiver,

stratoClient: StratoClient,

): ReadableStore[TweetId, SimClustersEmbedding] = {

ObservedReadableStore(

StratoFetchableStore

.withView[Long, SimClustersEmbeddingView, ThriftSimClustersEmbedding](

stratoClient,

SimclustersTweetColPath,

SimClustersEmbeddingView(

EmbeddingType.LogFavLongestL2EmbeddingTweet,

ModelVersion.Model20m145k2020))

.mapValues(SimClustersEmbedding(\_)))(

statsReceiver.scope("rms\_tweet\_log\_fav\_longest\_l2\_store"))

}

@Provides

@Singleton

@Named(ModuleNames.RmsUserFavBasedProducerEmbeddingStore)

def providesRepresentationManagerUserFavBasedProducerEmbeddingStore(

statsReceiver: StatsReceiver,

stratoClient: StratoClient,

): ReadableStore[UserId, SimClustersEmbedding] = {

ObservedReadableStore(

StratoFetchableStore

.withView[Long, SimClustersEmbeddingView, ThriftSimClustersEmbedding](

stratoClient,

SimclustersUserColPath,

SimClustersEmbeddingView(

EmbeddingType.FavBasedProducer,

ModelVersion.Model20m145k2020

)

)

.mapValues(SimClustersEmbedding(\_)))(

statsReceiver.scope("rms\_user\_fav\_based\_producer\_store"))

}

@Provides

@Singleton

@Named(ModuleNames.RmsUserLogFavInterestedInEmbeddingStore)

def providesRepresentationManagerUserLogFavConsumerEmbeddingStore(

statsReceiver: StatsReceiver,

stratoClient: StratoClient,

): ReadableStore[UserId, SimClustersEmbedding] = {

ObservedReadableStore(

StratoFetchableStore

.withView[Long, SimClustersEmbeddingView, ThriftSimClustersEmbedding](

stratoClient,

SimclustersUserColPath,

SimClustersEmbeddingView(

EmbeddingType.LogFavBasedUserInterestedIn,

ModelVersion.Model20m145k2020

)

)

.mapValues(SimClustersEmbedding(\_)))(

statsReceiver.scope("rms\_user\_log\_fav\_interestedin\_store"))

}

@Provides

@Singleton

@Named(ModuleNames.RmsUserFollowInterestedInEmbeddingStore)

def providesRepresentationManagerUserFollowInterestedInEmbeddingStore(

statsReceiver: StatsReceiver,

stratoClient: StratoClient,

): ReadableStore[UserId, SimClustersEmbedding] = {

ObservedReadableStore(

StratoFetchableStore

.withView[Long, SimClustersEmbeddingView, ThriftSimClustersEmbedding](

stratoClient,

SimclustersUserColPath,

SimClustersEmbeddingView(

EmbeddingType.FollowBasedUserInterestedIn,

ModelVersion.Model20m145k2020

)

)

.mapValues(SimClustersEmbedding(\_)))(

statsReceiver.scope("rms\_user\_follow\_interestedin\_store"))

}

}